Day 1- 113229047- Erukala Bharath Kumar

SDLC:

1 What is SDLC?

Software development life cycle is a structured process that is used to plan, design, develop, and test quality software. It is process which follows for building a software for organization.It has specific task to be performed at different stages in the process.

2 Why SDLC?

SDLC (Software Development Life Cycle) is essential because it provides a structured framework for creating high-quality software efficiently.

It helps organizations plan, design, develop, test, and maintain software systems in a systematic way. By following SDLC, teams can better manage resources, control costs, minimize risks, and ensure the final product meets user requirements.

The process also facilitates clear communication between stakeholders and provides documentation for future maintenance. Without SDLC, software development would be chaotic, prone to errors, and likely to exceed budgets and timelines.

3 Stages of SDLC?

SDLC involves 6 stages

1. Plan and requirement analysis
2. Defining
3. Designing
4. Building
5. Testing
6. Deployment and maintenance

4 SDLC models :

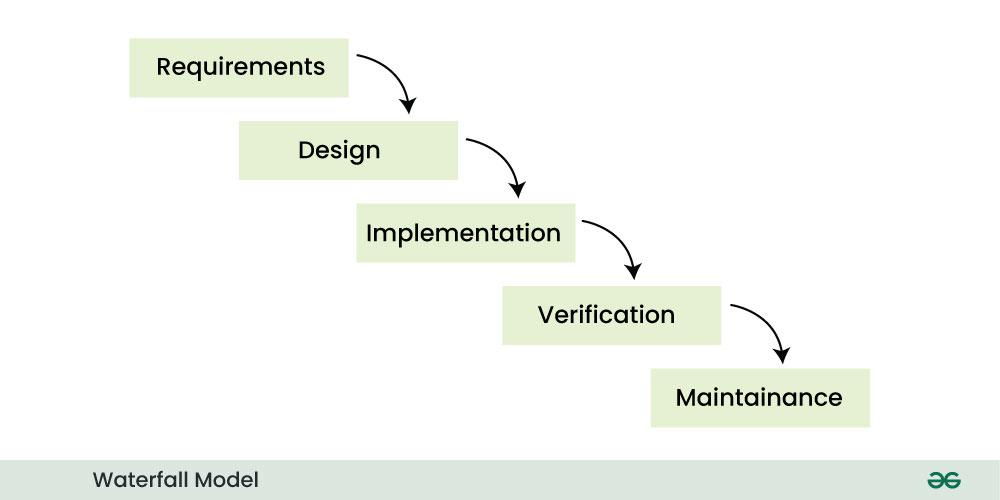
1. Waterfall Model:

- Sequential linear approach where each phase must complete before the next begins

- Phases include requirements, design, implementation, verification, and maintenance

- Best for projects with clear, unchanging requirements

- Easy to understand and manage but lacks flexibility for changes



2. Agile Model:

- Iterative approach with short development cycles (sprints)

- Emphasizes customer collaboration, adaptability to change, and frequent delivery

- Uses frameworks like Scrum, Kanban, or XP for implementation

- Ideal for projects with evolving requirements and need for quick feedback



3. Spiral Model:

- Risk-driven approach combining iterative development with waterfall's systematic aspects

- Each spiral cycle includes planning, risk analysis, engineering, and evaluation

- Emphasizes risk assessment throughout the development process

- Good for large, complex projects with significant risk factors



4. V-Model:

- Extension of waterfall with corresponding testing phases for each development phase

- Strong verification and validation focus with testing plans developed in parallel

- Each development stage has an associated testing phase

- Suitable for projects requiring high reliability and extensive testing



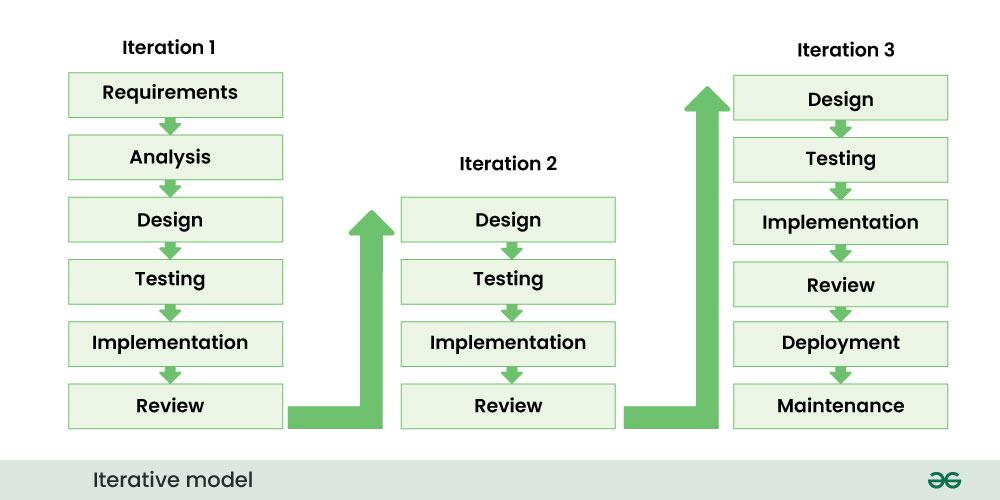
5.The Iterative model:

- Is about repeated cycles/iterations

- Builds small portions of software in each iteration

- Allows for refinement based on feedback

- Is good for projects where requirements may evolve



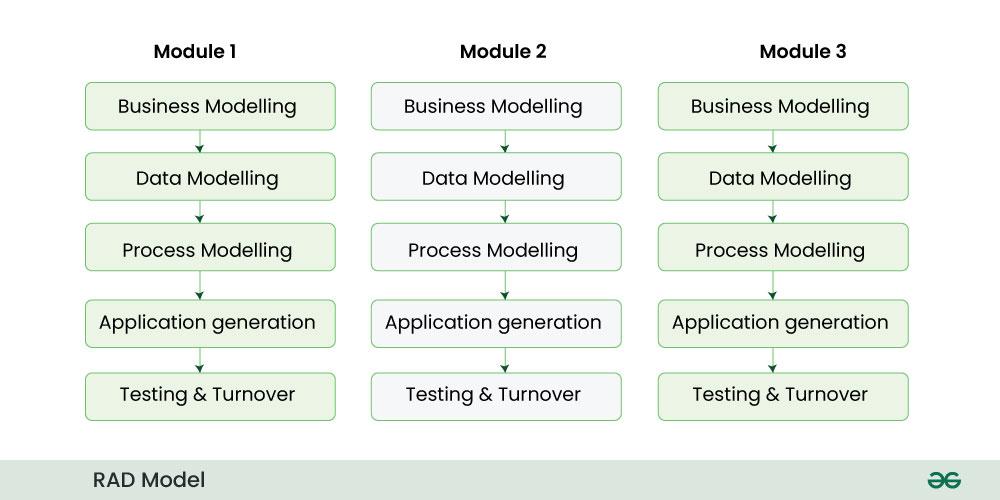
6. RAD (Rapid Application Development):

- Minimizes planning in favor of rapid prototyping and iterative development

- Uses workshops and focus groups to gather requirements

- Relies heavily on prototyping and reusable components

- Good for projects with tight timelines and flexible requirements



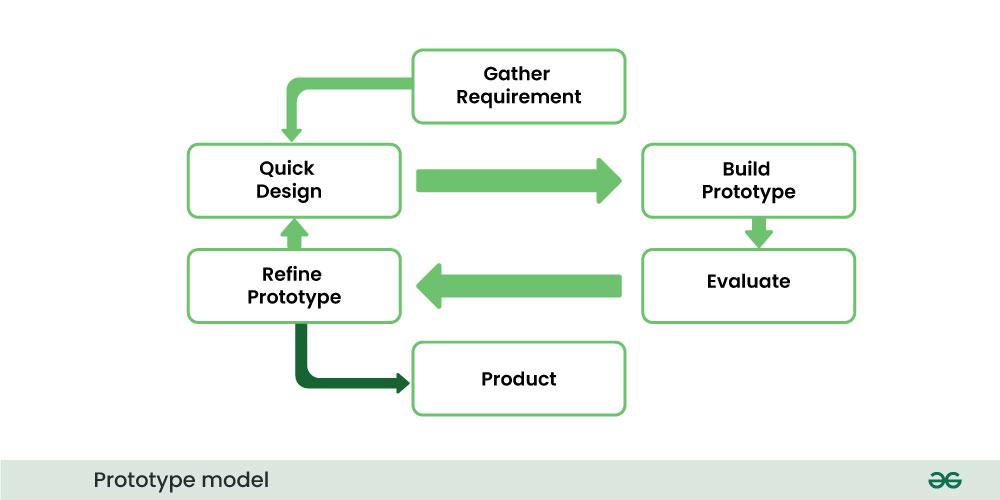
7. Prototype Model:

- Develops an initial version for user evaluation before full development

- Helps clarify uncertain requirements through user feedback

- Involves creating, testing, and refining prototypes until satisfactory

- Useful when requirements are not well understood initially



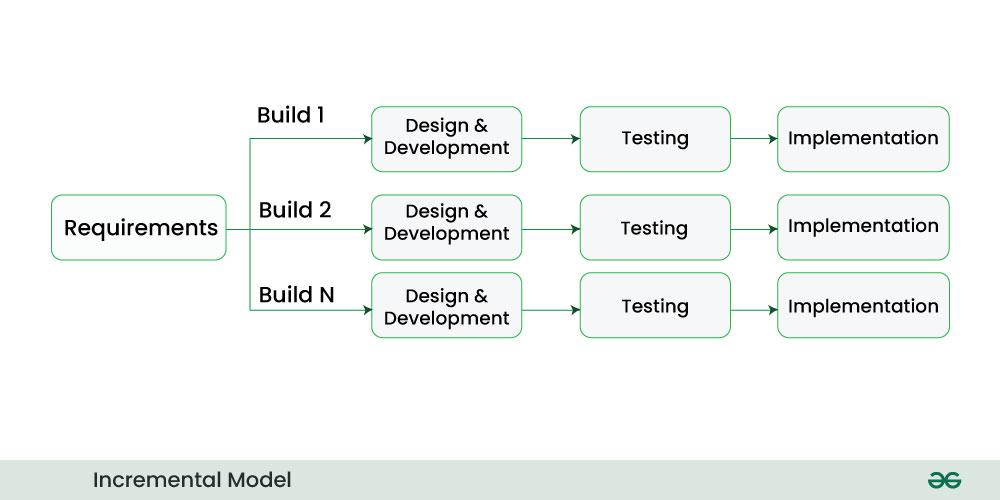
8. Incremental Model:

- Divides the product into builds with increasing functionality

- Delivers the product in pieces rather than all at once

- Each increment adds features to previously delivered increments

- Beneficial when staffing is limited or core features need early delivery



5 types of networks

1. LAN (Local Area Network)
2. WAN (Wide Area Network)
3. MAN (Metropolitan Area Network)
4. WLAN (Wireless Local Area Network)
5. VPN (Virtual Private Network)
6. VLAN (Virtual Local Area Network)

6 What are the types of servers?

1. Web Servers: These servers host websites and deliver web content to users when requested through browsers. Popular examples include Apache, Nginx, and Microsoft IIS.

2. Application Servers: These servers run and host applications and business logic that can be accessed by clients. They provide middleware services and support for developing and deploying applications (examples: Tomcat, JBoss, WebSphere).

3. Database Servers: These servers store, manage, and provide access to databases. They process database queries and manage database operations while ensuring data integrity and security (examples: MySQL, Oracle, SQL Server, MongoDB).

4. DNS Servers: Domain Name System servers translate human-readable domain names (like google.com) into IP addresses that computers can understand. They are essential for internet navigation and service discovery.

5. DHCP Servers: Dynamic Host Configuration Protocol servers automatically assign IP addresses and network configuration parameters to devices on a network. They simplify network administration by centralizing IP management.

6. FTP Servers: File Transfer Protocol servers enable the transfer of files between computers on a network. They provide authenticated access for uploading, downloading, and managing files remotely.

7 What do you know about DNS? Domain Name

DNS (Domain Name System) is the internet's address book, translating human-friendly domain names (like google.com) into machine-readable IP addresses (like 142.250.190.78). It operates as a hierarchical, distributed database system spread across numerous servers worldwide. The DNS resolution process works when you enter a website address: your computer queries a DNS resolver, which then navigates through the DNS hierarchy (root servers, TLD servers, and authoritative nameservers) to find the correct IP address. This entire process typically happens in milliseconds. DNS also enables email routing, load balancing, and service discovery, making it a critical foundation of virtually all internet and network services.

8 What is TCP and UDP? What is the difference?

TCP (transmission Control protocol) A connection-oriented protocol that guarantees reliable, ordered, and error-checked delivery of data between applications. TCP establishes connections before transmission, acknowledges received packets, and retransmits lost data, ensuring complete and accurate delivery at the cost of some overhead

UDP (User Datagram Protocol) A connectionless protocol that sends data without establishing connections, verifying delivery, or ensuring packet order. UDP offers minimal overhead and faster transmission by omitting error recovery and flow control mechanisms, making it ideal for applications where speed is more important than reliability.

Key differences between TCP and UDP:

1. Connection:

- TCP: Connection-oriented protocol that establishes a connection before data transfer

- UDP: Connectionless protocol that sends data without establishing a connection first

2. Reliability:

- TCP: Reliable delivery with acknowledgments, retransmission of lost packets, and error checking

- UDP: Unreliable delivery with no guarantees, no retransmission of lost packets

3. Ordering:

- TCP: Maintains packet order, delivers data to the application in the same order it was sent

- UDP: No packet ordering, packets may arrive in any order

4. Speed/Overhead:

- TCP: More overhead due to connection management, error checking, and flow control

- UDP: Less overhead, faster transmission with minimal protocol mechanisms

5. Use cases:

- TCP: Used for applications requiring reliable data delivery (web browsing, email, file transfers)

- UDP: Used for speed-sensitive applications where some data loss is acceptable (video streaming,

on line gaming, VoIP)

6. Header size:

- TCP: 20-60 bytes header

- UDP: 8 bytes header (smaller and more efficient)

7. Flow control:

- TCP: Has congestion control and flow control mechanisms

- UDP: No congestion control or flow control

9 What do you know about mac address? What is the difference between Mac address and IP address?

MAC Address (Media Access Control) A hardware-level address permanently assigned to a network interface card by the manufacturer. It's a 48-bit identifier (like 00:1A:2B:3C:4D:5E) that functions at the Data Link Layer (Layer 2) of the OSI model. MAC addresses uniquely identify physical devices on a local network segment and don't change when moving between networks.

IP Address A logical address assigned to a device on a network that functions at the Network Layer (Layer 3). IP addresses can be dynamic (changing) or static, exist in formats like IPv4 (192.168.1.1) or IPv6, and are used for routing traffic across different networks. While MAC addresses identify the physical device, IP addresses identify its location on a network topology.

Key Difference MAC addresses identify the physical hardware and work only on local networks, while IP addresses are logical identifiers that enable routing between different networks and can change as devices move between networks.

10 What is OSI Model

The OSI (Open Systems Interconnection) Model is a conceptual framework that standardizes the functions of a communication system into seven distinct layers. it provides a common language for network communications.

The seven layers, from bottom to top, are:

Physical Layer: Handles raw bit transmission over physical media (cables, hardware)

Data Link Layer: Manages node-to-node communication and error detection (MAC addresses, switches)

Network Layer: Determines paths and logical addressing (IP addresses, routers)

Transport Layer: Transmit data and manages end-to-end connections and reliability (TCP/UDP, ports)

Session Layer: Establishes, manages, and terminates connections

Presentation Layer: Translates data between the application and network formats (encryption, formatting)

Application Layer: Provides network services directly to end-users (HTTP, FTP, SMTP)



11 IPv4 Address and Classes

An IPv4 address is a 32-bit numerical label that identifies a device on a computer network using the Internet Protocol. It appears as four numbers separated by periods (e.g., 192.168.1.1), with each number ranging from 0 to 255. IPv4 addresses serve as unique identifiers that allow devices to communicate over IP networks, functioning like digital postal addresses for routing data packets to their correct destinations. With approximately 4.3 billion possible unique addresses.

Class A (1-126.x.x.x) First bit is 0. Designed for huge networks with 16+ million hosts per network. Default subnet mask is 255.0.0.0.

Class B (128-191.x.x.x) First bits are 10. Supports medium-sized networks with up to 65,534 hosts each. Default subnet mask is 255.255.0.0.

Class C (192-223.x.x.x) First bits are 110. Used for small networks with up to 254 hosts each. Default subnet mask is 255.255.255.0.

Class D (224-239.x.x.x) Used for multicast groups, not individual hosts.

Class E (240-255.x.x.x) Reserved for experimental purposes, not generally used in public networks.

12 Advantages of Using a VPN

* Encrypts internet traffic, protecting sensitive data from hackers and surveillance.
* Masks your IP address, providing anonymity and preventing tracking.
* Bypasses geographic restrictions to access region-locked content.
* Creates a secure connection on public Wi-Fi networks, preventing data interception.

13 Types of VPN

Access VPN: Enables individual remote users to securely connect to a private network from any location using public infrastructure.

Site-to-Site VPN: Connects entire networks across multiple locations by creating encrypted tunnels between gateways at each site

Intranet VPN: Links multiple internal sites within the same organization, allowing employees to securely share resources across different office locations.

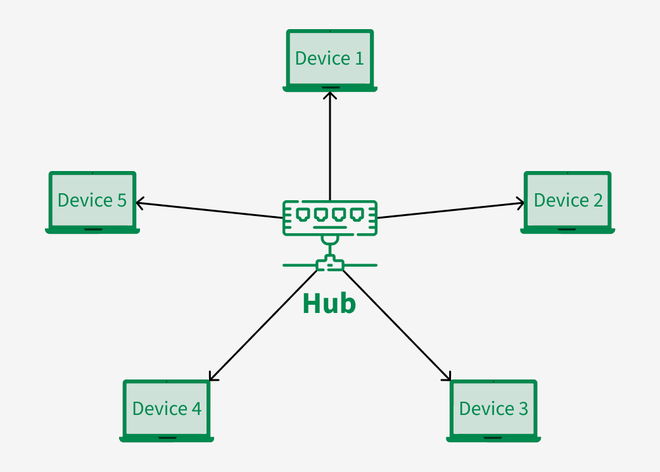
Extranet VPN: Extends secure network access to trusted external partners, suppliers, or customers while restricting them to specific resources.

14 Types of topologies:

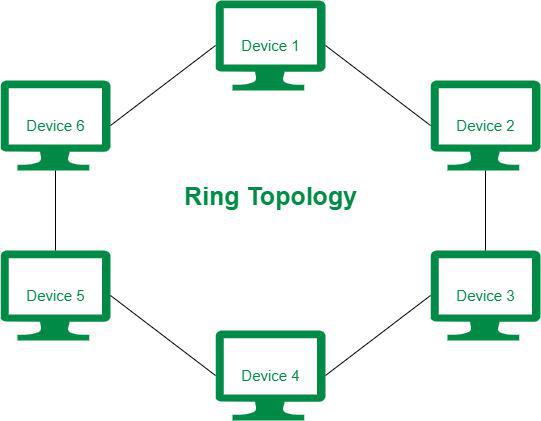
Bus Topology: All devices connect to a single central cable (backbone).



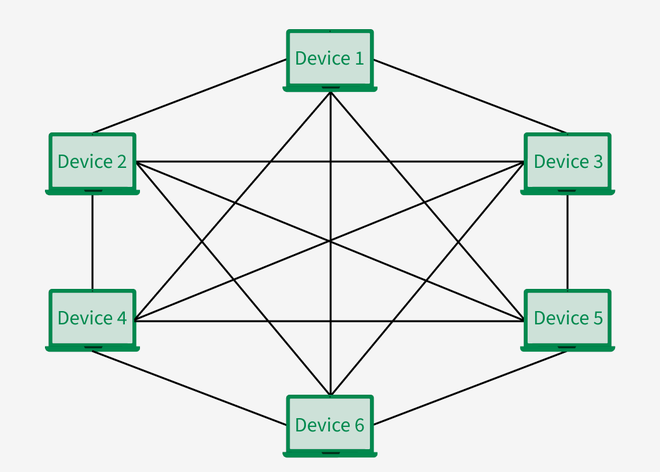
Star Topology: All devices connect to a central hub or switch.



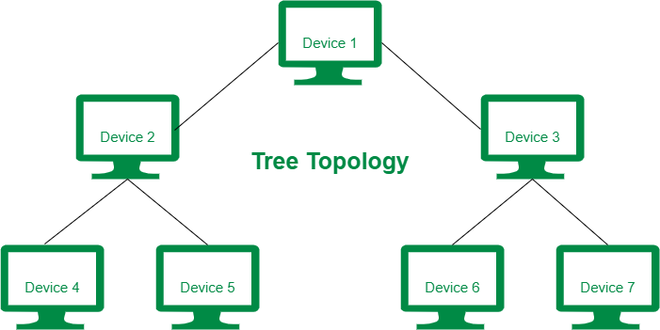
Ring Topology: Devices connect in a closed loop, with each device linked to two others.



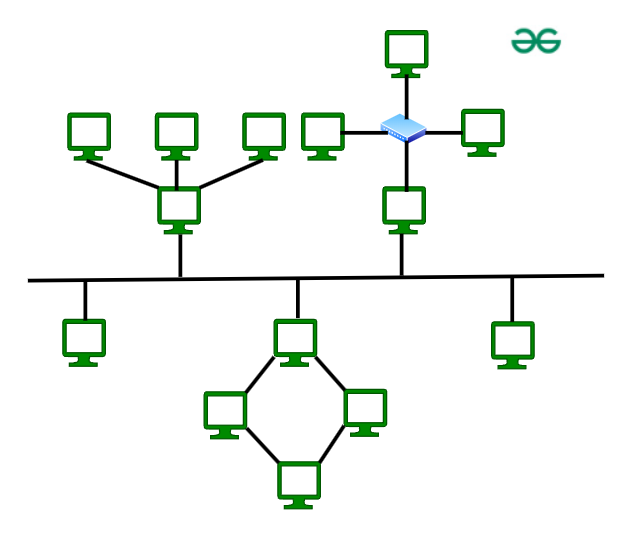
Mesh Topology: Devices interconnect with multiple direct links to other devices.



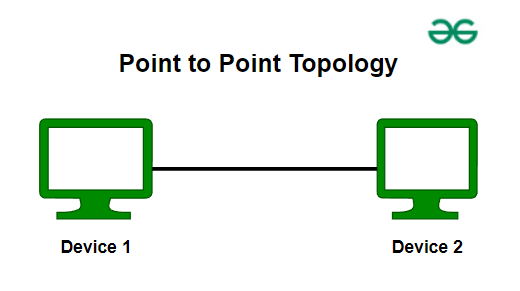
Tree Topology: Hierarchical structure combining elements of bus and star topologies.



Hybrid Topology: Combination of two or more different topology types.



Point-to-Point Topology: Direct connection between exactly two network nodes.



15 Extended Bus Topology

Extended bus topology is a variation of the standard bus topology where the main backbone cable is extended using repeaters or amplifiers to cover larger areas. It overcomes distance limitations of traditional bus networks by connecting multiple cable segments together, allowing signals to travel further. This topology may include branches that split off from the main bus, creating additional paths for devices to connect while maintaining the fundamental characteristics of a bus structure. While it enables greater network coverage, it still shares the standard bus topology's vulnerability to single cable failures.

16 What is the use of a router and how is it different from a gateway?

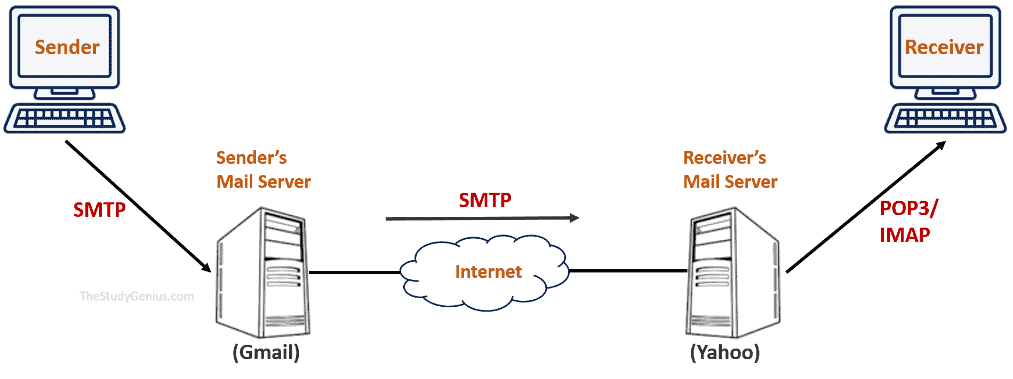
A router is a network device that forwards data packets between computer networks based on IP addresses. It connects devices within a local network (LAN) and connects that network to other networks, including the internet. Routers make intelligent decisions about the best path for data to travel using routing tables, control network traffic, provide firewall protection, and often offer wireless connectivity. They primarily operate at the Network layer (Layer 3) of the OSI model.

A gateway is a network point that acts as an entrance to another network, serving as a "gate" between different network environments. It translates between different network protocols, architectures, or data formats, enabling communication between otherwise incompatible systems. Gateways can work across multiple OSI layers and perform more complex translations than routers.

Key Difference:

While routers focus on connecting similar networks and routing packets based on IP addresses, gateways serve as translators between different network types, protocols, or environments. All routers can function as gateways, but not all gateways are routers. Gateways have broader functionality in bridging fundamentally different network systems.

17 SMTP



17 OSI vs TCP/IP Models

OSI Model has seven layers (Physical, Data Link, Network, Transport, Session, Presentation, Application) and serves primarily as a theoretical reference framework. Developed by ISO, it provides clearly defined functions for each layer with strict boundaries between them.

TCP/IP Model has only four layers (Network Interface, Internet, Transport, Application), combining several OSI layers. It was developed practically for internet communication and is the actual implementation used in real-world networking. Unlike the conceptual OSI model, TCP/IP was built around specific protocols and evolved from practical networking needs.

| **Parameters** | **OSI Model** | **TCP/IP Model** |
| --- | --- | --- |
| **Full Form** | OSI stands for Open Systems Interconnection | TCP/IP stands for Transmission Control Protocol/Internet Protocol |
| **Layers** | It has 7 layers | It has 4 layers |
| **Usage** | It is low in usage | It is mostly used |
| **Approach** | It is vertically approached | It is horizontally approached |
| **Delivery** | Delivery of the package is guaranteed in OSI Model | Delivery of the package is not guaranteed in TCP/IP Model |
| **Replacement** | Replacement of tools and changes can easily be done in this model | Replacing the tools is not easy as it is in OSI Model |
| **Reliability** | It is less reliable than TCP/IP Model | It is more reliable than OSI Model |
| **Protocol Example** | Not tied to specific protocols, but examples include HTTP (Application), SSL/TLS (Presentation), TCP (Transport), IP (Network), Ethernet (Data Link) | HTTP, FTP, TCP, UDP, IP, Ethernet |
| **Error Handling** | Built into Data Link and Transport layers | Built into protocols like TCP |
| **Connection Orientation** | Both connection-oriented (TCP) and connectionless (UDP) protocols are covered at the Transport layer | TCP (connection-oriented), UDP (connectionless |

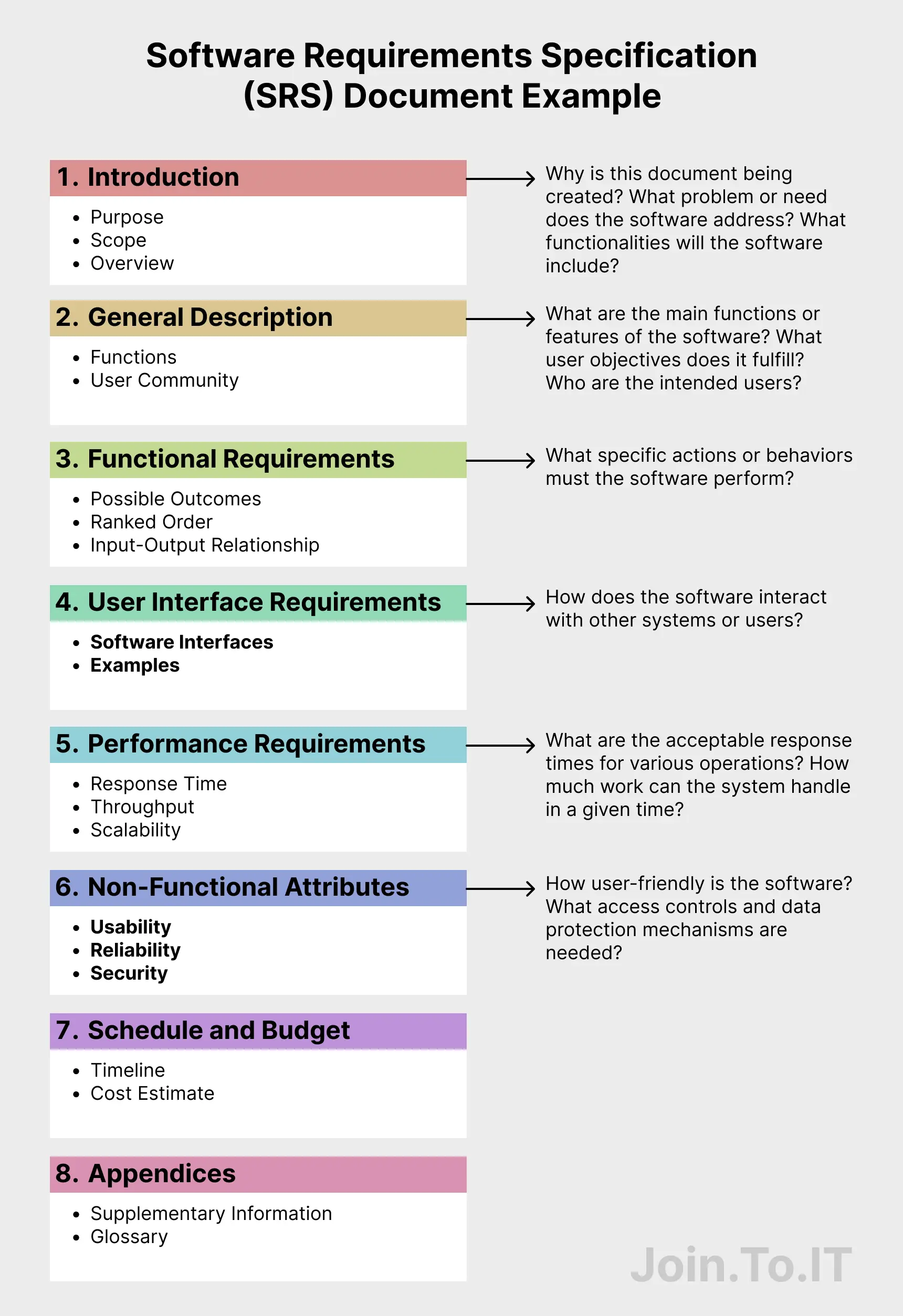
18 HLD and LLD in SDLC

High-Level Design (HLD): Occurs early in SDLC after requirements gathering, focusing on the system's overall architecture, major components, and their interactions. HLD establishes the technical framework without diving into implementation details, helping stakeholders understand the solution approach.

Low-Level Design (LLD): Follows HLD in the development process, providing detailed specifications for implementation including class structures, algorithms, and database schemas. LLD serves as the direct blueprint for developers to begin coding with precise technical instructions.

19 Software Requirements Specification (SRS)

A Software Requirements Specification (SRS) is a comprehensive document that defines all functional and non-functional requirements for a software project. It serves as a contract between stakeholders and development teams by clearly describing what the software must do rather than how it will do it. An SRS typically includes system purpose, features, interfaces, constraints, and performance requirements, providing a foundation for design, development, testing, and project management activities.



**SDLC MCQ**

1.

A feasibility study using the SDLC model is conducted to

determine whether or not the project is technically possible

determine whether the proposal is financially viable

Both a and b

None of the above

2.

A well-documented life cycle model aids in the detection of what during the development phase?

Inconsistencies

Redundancies

Omission

All of the above

3.

How many lines of code does the Build & Fix Model suit for programming exercises?

100-200

300-400

600-700

Above 800+

4.

In which life cycle does regression testing play a significant role?

Waterfall model

V model

Iterative model

All of the above

5.

What determines if the project should go forward?

feasibility assessment

opportunity identification

system evaluation

program specification

6.

What is the most significant disadvantage of employing the RAD Model?

Developers/designers that are highly specialized and skilled are required.

Component reusability is improved.

Encourages client/customer input.

Increases component reusability.

7.

Which of the following developmental models is incremental?

Prototyping, V model, Agile

Prototyping, RAD, Agile, RUP

Prototyping, V model, RAD, Agile, RUP

All of the above

8.

Which of the following is an Agile development characteristic?

Shared code ownership

Test-Driven Development

Implement the simplest solution to meet today's problem

Continual feedback from customer

All of the above

9.

Which of the following steps in the SDLC framework are valid?

Requirement Gathering

Software Design

System Analysis

All of the above

10.

Who is in charge of system development, staffing, budgeting, and reporting, as well as ensuring that deadlines are met?

Project managers

Network engineers

Graphic designers

Systems analysts